

Programa del curso IC-1802

## **Introducción a la Programación**

**Escuela de Computación**  
**Carrera de Ingeniería en Computación, Plan 410.**

## I parte: Aspectos relativos al plan de estudios

### 1 Datos generales

<b>Nombre del curso:</b>	Introducción a la Programación
<b>Código:</b>	IC-1802
<b>Tipo de curso:</b>	Teórico-Práctico
<b>Electivo o no:</b>	No
<b>Nº de créditos:</b>	3
<b>Nº horas de clase por semana:</b>	4
<b>Nº horas extraclasses por semana:</b>	5
<b>Ubicación en el plan de estudios:</b>	Curso del 1er Semestre de la carrera de Ingeniería en Computación
<b>Requisitos:</b>	Ninguno
<b>Correquisitos:</b>	IC-1803 Taller de Programación
<b>El curso es requisito de:</b>	IC-2101 Programación Orientada a Objetos
<b>Asistencia:</b>	Obligatoria
<b>Suficiencia:</b>	No
<b>Posibilidad de reconocimiento:</b>	Sí
<b>Vigencia del programa:</b>	II Semestre de 2021

## 2 Descripción general

Este curso comprende una introducción a los conceptos básicos para modelar y solucionar problemas de forma algorítmica, así como las estrategias fundamentales para la construcción de programas de computadoras.

## 3 Objetivos

### Objetivo General

Al finalizar el curso el estudiante debe ser capaz de desarrollar algoritmos básicos y realizarlos de manera sistemática en un modelo computacional específico o lenguaje de programación.

### Objetivos Específicos

1. Conocer algoritmos fundamentales de las ciencias de la computación.
2. Especificar algoritmos por medio de abstracciones funcionales.
3. Conocer y delimitar las etapas básicas asociadas en la construcción de algoritmos.
4. Analizar la calidad, correctitud, eficiencia y mantenibilidad de una solución algorítmica.
5. Conocer y utilizar los principales elementos semánticos y sintácticos de un modelo de programación concreto.

## 4 Contenidos

### 1. Resolución de problemas

- 1.1. Algoritmos
- 1.2. Corrida ejemplo
- 1.3. Definición del problema
- 1.4. Diseño del algoritmo
- 1.5. Entradas, salidas, proceso y limitaciones
- 1.6. Verificación y análisis del algoritmo
- 1.7. Validación del programa

### 2. Técnicas con listas

- 2.1. Concepto de lista
- 2.2. Pertenencia, unión, intersección, etc.
- 2.3. Eliminar duplicados

- 2.4. Búsqueda
- 2.5. Ordenamientos

- 2.6. Relaciones y composiciones

### **3. Algoritmos numéricos**

- 3.1. Funciones matemáticas básicas

- 3.2. Series

- 3.3. Sucesiones

- 3.4. Métodos numéricos

- 3.5. Aritmética modular, números primos

### **4. Estructuras avanzadas**

- 4.1. Árboles y su manipulación

- 4.2. Árboles binarios ordenados

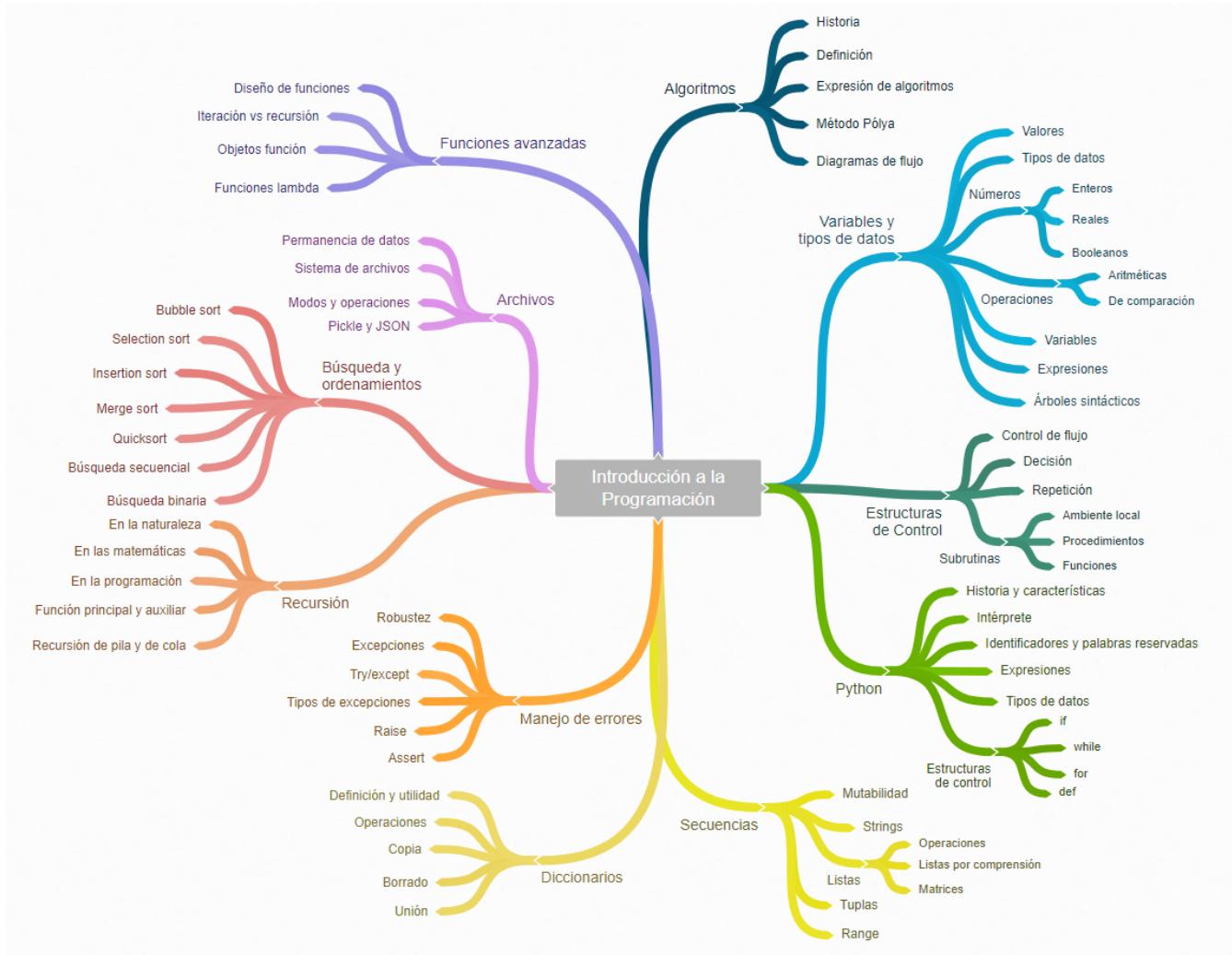
- 4.3. Variables imperativas

- 4.4. Iteración

- 4.5. Funciones como datos

- 4.6. Manejo imperativo de datos

- 4.7. Manejo básico de objetos



## II parte: Aspectos operativos

### 5 Metodología de enseñanza y aprendizaje

Se emplearán técnicas de clases magistrales por parte del profesor, en donde se desarrollarán los aspectos teóricos y prácticos más relevantes de los diferentes temas. Además, se combinarán con una alta participación por parte de los estudiantes durante el transcurso de las lecciones, por medio de llamadas orales, respuestas a casos en la pizarra y de trabajos en grupo.

Se presupone que el alumno profundiza los temas abordados en la clase en las lecturas recomendadas por el profesor y que el estudiante será responsable de desarrollar los proyectos programados que se asignen en el curso.

### 6 Evaluación

El curso será evaluado mediante dos rubros principales: exámenes parciales y exámenes cortos.

#### Exámenes parciales (60%):

Se realizarán tres exámenes parciales a través del semestre.

#### Exámenes cortos (20%):

Durante el semestre se realizarán un total de 10 exámenes cortos (quices) que evalúan temas puntuales recién vistos.

Rubro	Porcentaje
1er examen parcial	20%
2do examen parcial	20%
3er examen parcial	20%
Exámenes cortos	20%
Bitácora de apuntes	20%
<u>Total</u>	100%

#### Cronograma de Actividades

Actividad	Semanas
Resolución de problemas	4
Técnicas con listas	4
Algoritmos numéricos	4
Estructuras avanzadas	4

## 7 Bibliografía

Se utilizarán los siguientes textos:

- Brookshead, J. Glenn. Computer Science, an Overview.  
Lutz, M. Learning Python. 5th ed., O'Reilly, 2013.  
Lutz, M. Programming Python. 4th ed., O'Reilly. 2011.

## 8 Profesor



Mauricio Avilés Cisneros tiene estudios de Ingeniería en Computación (ITCR) y Maestría en Educación con énfasis en Docencia (UAM). Ha laborado para la institución desde 2003, primero como analista de sistemas del Departamento de Administración de Tecnologías de Información y Comunicación (DATIC), profesor de los cursos Computación para Administración y Sistemas de Información para Administración de la escuela de Administración de Empresas durante el periodo 2007-2011 y 2018 en adelante. Actualmente, labora como profesor e investigador a tiempo completo en la Escuela de Computación desde 2012, donde ha impartido los cursos de Introducción a la Programación, Taller de Programación, Estructuras de Datos, Programación Orientada a Objetos, Análisis y Diseño de Algoritmos, y Simulación de Sistemas Naturales.

Ubicación: Centro Académico de San José.

Teléfono oficina: 25509586

Correo electrónico: [maviles@tec.ac.cr](mailto:maviles@tec.ac.cr)

Medio oficial electrónico: TEC-Digital (<http://tecdigital.tec.ac.cr>)

Consulta: Se realizará con cita previa, con horario y lugar a convenir según el profesor y el estudiante.

Horario de consulta:

Martes	10:00-12:00
Miércoles	13:00-15:00
Jueves	10:00-12:00
Viernes	13:00-15:00

Adicionalmente se brinda consulta mediante redes sociales y correo electrónico, en modalidad 24/7, limitado por las posibilidades del profesor.

**9 Asistente**



María Fernanda Niño, estudiante avanzada de Ing. en Computación (San José). La mayoría del tiempo está disponible en Discord para cualquier consulta o duda que tengan.

Teléfono: 89952503

Correo: [maf.nino7@gmail.com](mailto:maf.nino7@gmail.com)